ABV- A Low Speed Automation Project to Study the Technical Feasibility of Fully Automated Driving

Paulo Resende, Evangeline Pollard, Hao Li, Fawzi Nashashibi

Abstract— **The purpose of the ABV project was to demonstrate the technical feasibility of fully automated driving at speeds below 50 km/h in urban and suburban areas with adequate infrastructure quality (no intersections, known road geometry and lane markings available). Researchers of Inria were in charge of the automation of an electrified Citröen C1 Ev'ie. The goal of this article is to present the results of the ABV project and to draw new perspectives for urban autonomous driving.**

The Intelligent Transportation Systems (ITS) community has been focusing on vehicle automation since many years. From 1987 to 1995 the European Commission funded the 800 million Euros EC EUREKA Prometheus Project on autonomous vehicles where vehicles such as the Dickmann's Mercedes-Benz vehicle were able to perform vision-based autonomous navigation on empty roads. In the meanwhile, the USA funded the DARPA Autonomous Land Vehicle (ALV) project. Here, laser-based (ERIM) and vision-based navigation (CMU, SRI) were developed for low speed driving on roads while HRL laboratories were demonstrating the first off-road low speed autonomous ALV robot. In the 90's, the cybercar were one of the first concepts of vehicles designed as fully automated vehicles [1]. Beyond its fully automation ability, they were conceived as a new transportation system, for passengers or goods, on a road network with on-demand and door-to-door capability. In the USA, the Federal Highway Administration established the National Automated Highway System Consortium (NAHSC) that demonstrated about 20 automated vehicles in Demo'97 on I-15 in San Diego, California.

Since then, large efforts have been performed towards full automation with the first DARPA Grand Challenge in 2004 [2] and its urban version in 2007 (DARPA Urban Challenge [3]). In 2010, the VisLab Intercontinental Autonomous Challenge, consisted in riding four vehicles from Parma, Italy, to Shanghai, China, mainly in an automated way [4]. The same year, Google announced having created an autopilot system for cars which already drove more than 200,000 km in a fully automated way [5]. However, the technology used is really expensive. Other initiatives and automated vehicles exist worldwide *e.g.* Technical University of Braunschweig or Stanford's road vehicles.

The Google driverless car is the first case of a vehicle supposed to autonomously drive in a urban context. Even if Nevada was the first state to issue driverless vehicle licenses on public roadways, total automation remains a future goal, even if vehicles become more and more autonomous in the automotive industry. Indeed, many Advanced Driver Assistance Systems (ADAS) are now embedded into industrial cars in order to help the driver in his driving process. Part of these systems, such as speed alert or blind spot detection, provides only advice or warning to the driver, but others, such as the Adaptive Cruise Control (ACC) or the Anti-lock braking system (ABS), can be considered as elements of a partial automation since they act on the control part of the vehicle. This increasing automation is possible because sensors become more and more effective and reliable. In this spirit, many international and European projects aim at proving the technical feasibility for such automated vehicles in order to create a legal framework. That was the case with the Citymobil project, the German AKTIVE project and the HAVEit project [6].

Beyond the HAVEit project, that was related to high speed automation on highways and human machine cooperation, the French ABV project, standing for Low Speed Automation, differs from HAVEit by focusing on congested and heavy traffic in urban and suburban roads at speeds below 50 km/h and adding the fully automated driving capability to the automation spectrum.

By automatically following congested traffic, the ABV system relieves the human driver from performing monotonous tasks like holding the brake pedal or rather risky maneuvers like changing lanes or simply keeping a safe distance from the vehicle in front. During fully automated driving inside the application zone, the human driver is still responsible for the vehicle, and so, he is required to be involved in the driving task (e.g. activating a blinker to overtake) but with much less engagement. When reaching the end of the application zone the driver is required to take over the control of the vehicle. If he fails to do so, the vehicle will automatically stop.

To demonstrate the technical feasibility of the ABV project, technologies where integrated into a demonstration vehicle consisting of an electrified Citröen C1 Ev'ie [7].

Consequently, the paper is structured as follows. First, a system description is provided in Sec. I. Then, from Sec. II to III, the three classical components of automation, perception-planing-control, are described. Sec. IV is dedicated to the driving share. And, in Sec. V, we report results of our scenarios experiments. Finally, conclusion and future works are drawn in Sec. VI.

The automation algorithms, following the classical scheme (1) perception (2) planning (3) control illustrated in Fig. 1.

P. Resende, E. Pollard, H. Li and F. Nashashibi are with the INRIA institute (IMARA team). Domaine de Voluceau, B.P. 105, 78153 - Le Chesnay, France firstname.lastname@inria.fr

They have been integrated into an electrified Citröen C1 Ev'ie and tested on several experiments on the Satory tracks (Versailles, France) shown in Fig. 2.

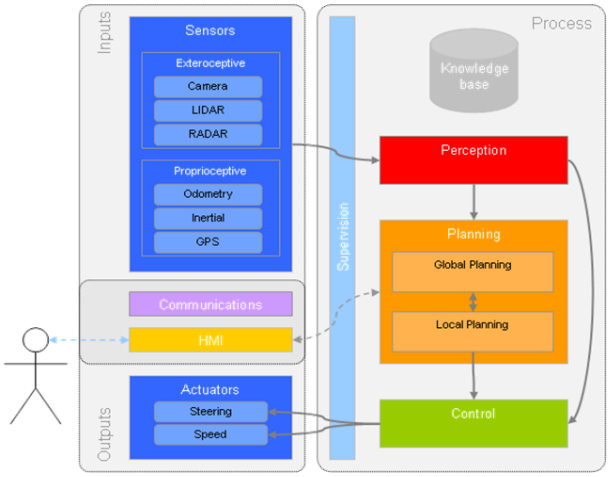

Fig. 1. Automation loop for ABV

Fig. 2. Satory tracks

I. System description

An overview of the vehicle equipment and components can be seen on the Fig. 3. Citröen C1 Ev'ie is an electrical vehicle equipped the following elements:

- Sensors in order to provide information about the surrounding environment (more information are provided in Sec. I-A) .
- On board computer to process data coming from the sensor in order to plan the trajectory and calculate forces to apply on the actuators to achieve it.
- Actuators in order to follow the planed trajectory
- Interfaces in order to interact with the driver through a touch screen and switches to activate the different sensor and the automation mode
- Communication devices: CAN bus (for Controller Area Network) for the low level communication and 3G antenna for the internet.

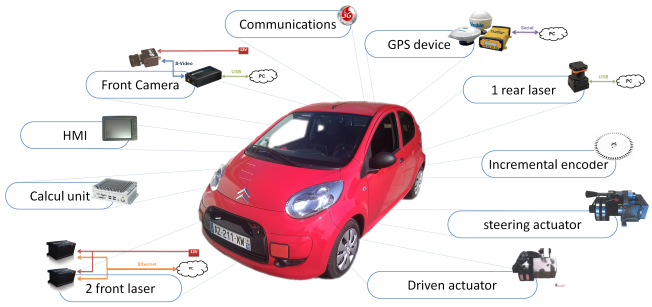

Fig. 3. Vehicle equipment

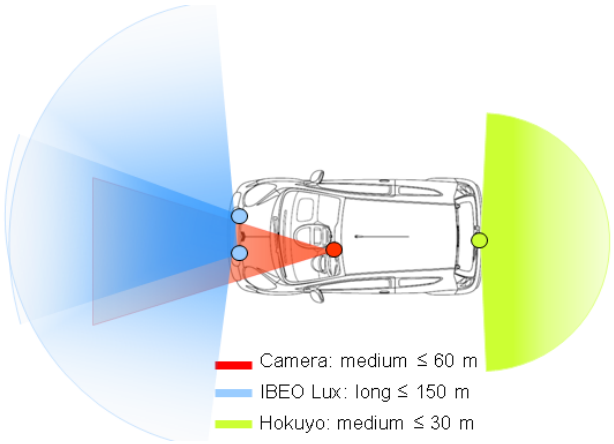

Fig. 4. Top-view of the perception area

A. Sensors

Exteroceptive sensors are used to provide information about the environment. Two laser range finder devices are mounted in the front of the vehicle and one at the back as described in Fig. 4. Two lasers are used at the front to cover all the front area because they have a 110 ° aperture only. One frontal camera is used to observe the road. One GPS-RTK is used to provide an absolute positioning, because no map-matching is considered in this application. In addition, proprioceptive sensors such as incremental encoders are used to provide the vehicle velocity and steering angle information are provided from the steering actuator.

B. Actuators

The vehicle is equipped with acceleration/brake and steering actuators (PARAVAN) that are controlled by software. Actuator controller and CAN bus architecture are described in Fig. 5.

II. Perception issues

Perception issues are divided into three main axes. First, the ego-localization consists in estimating the vehicle position and orientation in the application zone.

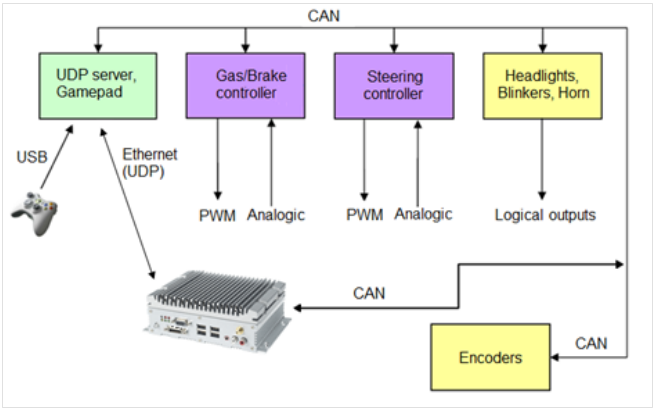

Fig. 5. Actuator controllers and CAN bus configuration

A. Ego-localization

Let the vehicle state be denoted as $X_t = (x_t, y_t, \theta_t)^T$; where (x_t, y_t) and θ_t respectively denote the position and the heading angle of the vehicle in the absolute reference at time t. The vehicle state is estimated by fusing GPS data and odometer data in the framework of the Extended Kalman Filter (EKF) [8].

Vehicle state evolution prediction

Let $u_t = (\Delta d_t, \Delta\theta_t)$ denote the vehicle motion data, *i.e.* the traversed distance and the yaw change (directly measured or indirectly computed). The vehicle state equation is thus written as:

$$\mathbf{X}_t = G(\mathbf{X}_{t-1}, \mathbf{u}_t, \mathbf{E_x}) \tag{1}$$

where G is the non linear motion model and $\mathbf{E_x}$ is a zero-mean, white Gaussian process, which models the uncertainty on the vehicle state model.

Motion data are used to predict the vehicle state $\hat{\mathbf{X}}_{t|t-1}$ according to the kinematic bicycle model:

$$\begin{cases} x_{t|t-1} \approx x_{t-1} + \Delta d_t \cos(\theta_{t-1} + \Delta\theta_t/2) \\ y_{t|t-1} \approx y_{t-1} + \Delta d_t \sin(\theta_{t-1} + \Delta\theta_t/2) \\ \theta_{t|t-1} = \theta_{t-1} + \Delta\theta_t \end{cases} \tag{2}$$

The predicted vehicle state covariance $\mathbf{P}_{t|t-1}$ is evolved as:

$$\mathbf{P}_{t|t-1} = \mathbf{G_X} \mathbf{P}_{t-1} \mathbf{G_X}^T + \mathbf{G_u} \mathbf{E_u} \mathbf{G_u}^T \tag{3}$$

where $\mathbf{G_X}$ and $\mathbf{G_u}$ are respectively the Jacobian matrices of the function G with respect to the $\mathbf{X}$ and $\mathbf{u}$.

Vehicle state update

Let the GPS measurement for the vehicle at time t be denoted as $\mathbf{z_t} = (z_x, z_y)^T$. The measurement model can be described as:

$$\mathbf{z}_t = \mathbf{H}\mathbf{X}_t + \mathbf{R_z} \tag{4}$$

where $\mathbf{H} = [\mathbf{I}_{2\times2}, \mathbf{0}_{2\times1}]$ is the transition matrix from the vehicle state space to the measurement space; the measurement error $\mathbf{R_z}$ is a zero-mean, white Gaussian process with a known covariance matrix $\mathbf{E_z}$

The state update is carried out as follows:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{t|t-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{E_z})^{-1} \\ \hat{\mathbf{X}}_t &= \mathbf{X}_{t|t-1} + \mathbf{K}(\mathbf{z}_t - \mathbf{H}\mathbf{X}_{t|t-1}) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{t|t-1} \end{aligned} \tag{5}$$

where $\mathbf{K}$ is defined as the gain of the Kalman filter.

B. Lane detection and tracking

The lane detection process is fulfilled by a mono-vision system. It is divided into three main steps: lane marking segment extraction, lane model fitting, and lane tracking.

In the lane marking segment extraction, a bird-eye view images is generated by inverse perspective mapping *i.e.* projecting the camera image onto the ground plane. Camera parameters which determine the geometrical relationship between the image plane and the ground plane are calibrated off-line. Image processing is carried out on the bird-eye view image instead of directly on the camera image. Based on some criteria that are related to the inherent physical properties of the lane marks only, lane mark segment candidates are extracted using Canny detector for edge detection, edge point connection and edge segment matching.

The lane model fitting is performed to generate possible lane configuration hypotheses based on the extracted lane mark segments and *a priori* knowledge about the road geometry. The lane model is parameterized in the world reference (the ground plane) instead of in the image reference. For any pair of lane marking segments, a preliminary judgment on whether they are possibly on the same lane is performed, according to the geometry of a fitted cubic spline and normal road construction standards.

Then, a group of lane configuration samples is generated randomly in the spirit of the RANSAC method [9] and their fitness values are evaluated. The lane tracking is carried out in the framework of the particle filter, including the evolution and the update of the lane state.

More details concerning this sub-section can be referred to [10].

C. Obstacle detection and tracking

1) Principle: The proposed obstacle detection and tracking approach is a classical 6 step approach [11] illustrated in Fig. 6.

- In the **Data processing** step, distances coming from the several laser sensors are converted into (x, y, z) points in the local Cartesian coordinate system centered on the middle back of the chassis of the vehicle. They are then sorted depending on their angle from the coordinate system center.
- In the **Segmentation** step, a Recursive Line fitting algorithm is used as in [12] with parameter d_1 and d_2 for the maximum distances to the closest segment and between two successive segments respectively. Before applying the Recursive Line fitting algorithm on the point set, it is clustered. Each sorted point is recursively associated to an existing cluster by measuring its distance to the last point of the cluster or it is initializing a new one if

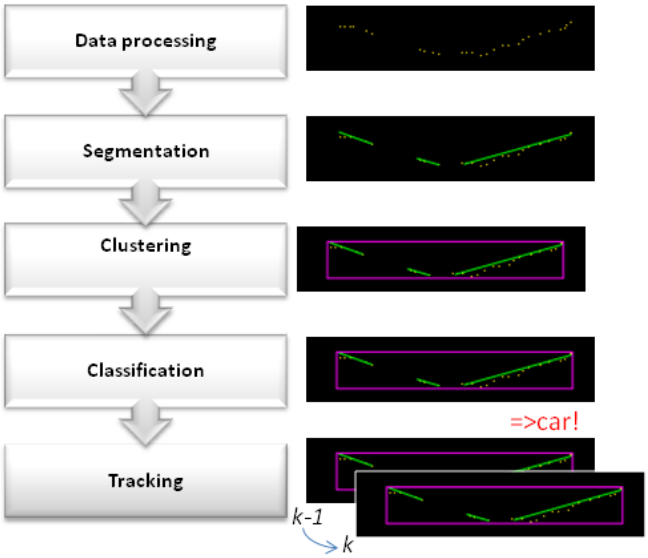

Fig. 6. Obstacle detection and tracking: general scheme

Fig. 7. Obstacle detection and tracking result - *Cyan rectangle is the ego-vehicle. Yellow points are the laser impacts. Green lines are the segments. Blue lines are the current lane limits. Pink lines are the detection area limits - Red square with label is one of the tracks and the associated red arrow is the estimated orientation*

the distance to each existing cluster is bigger than the same threshold d_2.

- In the **Clustering** step, segments are associated to create objects. Objects with less than 5 laser impacts are filtered. This number has been established by expertise. Then, only objects on the current lane are considered for ACC or emergency braking to limit false alarms.
- In the **Classification** step, size and shape consideration are used to obtain a raw classification of the object. This raw consideration is used to update the type probability over time only, but motion model is not yet considered.
- In the **Tracking** step, information about the vehicle dynamics are considered (velocity and steering angle) to improve the tracking of the object in the local Cartesian coordinate system. Object tracking is done in relative coordinates regarding the ego-vehicle using a Constant Velocity Kalman filter and Nearest Neighbor approach for data association to deal with real time constraint.

2) ACC Target detection with lane detection: In order to perform an ACC system, the closest obstacle on the current lane must be detected. Lane information (see Fig. 8) is thus converted into the local Cartesian coordinate system and used to filter obstacle on the current lane. As illustrated in Fig. 7, obstacles are tracked in the detection area (limits are the pink lines) and considered for the ACC only on the current lane (limits are the blue lines).

3) ACC Target Relative velocity estimation using numerical derivation method: In order to provide an ACC system, the relative velocity of the front obstacle must be calculated. This can be done by derivation of its relative distance.

a) Numerical derivation: The obstacle detection module provides an estimation of the relative position of the obstacles by regarding the ego position. From this position, the relative distance $d_r(t)$ of the closest front obstacle can be calculated over time. The distance $d_r(t)$ is thus considered as a noisy input signal to derive numerically, in order to gain the relative velocity.

Our approach is based on the Fliess's approach [13]. The relative distance $d_r(t)$ is approximated as a truncated Taylor expansion at order N, which can be locally approximated as a polynomial function. Here, for $t = 0$, we can approximate $d_r(t)$ as:

$$d_r(t) = \sum_{i \geq 0}^{N} d_r^{(i)}(0) \frac{t^i}{i!} \tag{6}$$

Then, each processed signal can be extended in a polynomial function of higher degree and derivative coefficient can be calculated by using the Laplace transform. Here, the relative distance $d_r(t)$ is locally represented as a first order polynomial function, $\forall (a_o, a_1) \in \mathbb{R}^2$:

$$d_r(t) = a_0 + a_1 \cdot t \tag{7}$$

In order to calculate the relative velocity, coefficient a_1 for $d_r(t)$ signal must be estimated. Using the Laplace transform and successive calculus transformation, eq (7) can be expressed in the Laplace domain as:

$$-\frac{a_1}{s^4} = \frac{D_r(s)}{s^2} + \frac{1}{s}\frac{dD_r(s)}{ds} \tag{8}$$

where $D_r(s)$ is the operational expression of $d_r(t)$. Using classical operational to time domain transformation rules and Cauchy formula, estimation of the coefficient $\hat{a}_1$ can be limited to the calculation of one integral:

$$\hat{a}_1 = -\frac{3!}{T^3} \int_0^T (T - 2\tau) d_r(\tau) d\tau \tag{9}$$

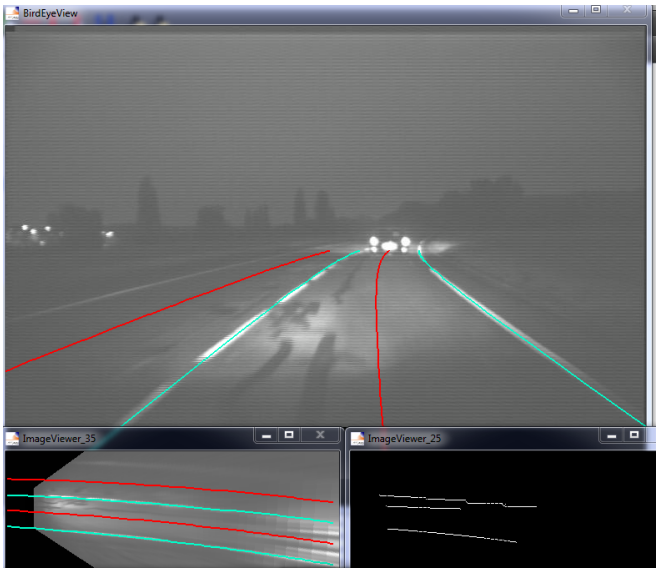

Fig. 8. Lane detection and tracking result - *Cyan lines are the current lane boundaries. Red lines are the lane centers. Bottom figures are bird eye view images: the right one is a binary image containing extraction points.*

where T is the length of the integration window. More details on this technique are provided in [14], [15].

b) A new robust numerical derivation method: The numerical derivation method has been extended in order to be robust to outliers. If for some reasons, the closest detected obstacle is not the right vehicle (in case of lane detection deficiency for instance, the guard rail can be detected instead of the front vehicle), then the value will be unexpected and has to not be taken into account for the relative velocity estimation.

The proposed robust approach can be summarized as follows. A maximum standard deviation σ is defined in order to define the tolerance of the system to outliers. It must be determined by considering the application and the maximum velocity and acceleration of the vehicle. Then for each incoming distance value $d_r(t)$ at time t, a gating test is processed which takes into account the time to the closest valid data Δt in order to balance the gating procedure over time. If the following condition described in eq. 10 is respected, then the incoming value is validated.

$$\frac{\left(d_r(t) - d_r(t-1)\right)^2}{\sigma^2} \cdot \exp(-\sigma \Delta t) < \gamma \tag{10}$$

where γ is coming from the χ^2 table following a gating probability of 0.99.

III. Control issues

The control is based on a decoupled control law algorithm: one for the steering wheels angle β (lateral control law Sec. III-A) to perform lane keeping/changing (LKS/LCS), and the other for the vehicle speed (longitudinal control law Sec. III-B) to perform an Adaptive Cruise Control (ACC).

A. Lateral control

The lateral control law for the steering wheel takes as inputs the lateral position error of the vehicle $e_\perp$, and the yaw angle of the vehicle Ψ at a variable look-ahead point. This point is located at a distance d_v, in front of the vehicle, depending on the vehicle actual speed v as shown in eq. (11).

$$d_v = \begin{cases} d_{min}, & v \leq v_{min} \\ d_{min} + \left(\dfrac{v_{max} - v}{v_{max} - v_{min}} \right), & v_{min} < v < v_{max} \\ d_{max}, & v \geq v_{max} \end{cases} \tag{11}$$

Eq. (12) represents the control law we used:

$$\beta = k_1 \cdot e_\perp + k_2 \cdot (\psi - \psi_{ref}) \tag{12}$$

In this equation the yaw angle reference ψ_{ref} is given by the target lane heading information. Proof of stability of this equation is given in [16].

The d_v function parameters together with the gains k_1 and k_2 are used to tune the system. The module of the maximum lateral position error and of the maximum steering angle, are respectively given by $e_{\perp,max}$ and β_{max}. The maximum steering wheels rate of the vehicle is respected by limiting the variation of β between two successive commands.

B. Longitudinal control

The longitudinal control law is a simple proportional-integral-derivative controller (PID) as shown on eq. (13) based on the speed error v_e given by $(v - v_{ref})$. This law is used to control the position p of a coupled gas/brake pedal.

$$p = k_p \cdot v_e + k_i \int (v_e + k_D) \partial v_e \tag{13}$$

To avoid saturation of the pedal position command the integral error is limited using an anti-windup incremental algorithm. When no obstacle is present in the trajectory of the vehicle, the reference speed v_{ref} is simply given by the minimum value between the driver preferred speed v_{driver}, the road speed limit v_{road}, and the maximum speed the automated system can handle v_{system}. When an obstacle is present in the trajectory of the vehicle moving at an absolute speed v_{obs}, the reference speed v_{ref} has to be calculated.

Considering the Torricelli's equation (with v_f and v_i the final and initial velocity),

$$v_f^2 = v_i^2 + 2A\Delta d \tag{14}$$

and knowing that the desired speed variation is zero when the ego-vehicle is at the appropriate distance to the front obstacle (*eg.* $d_{error} = 0$), the final velocity can be calculated as:

$$v_f = \sqrt{2.A.\Delta_d} \tag{15}$$

where $\Delta_d = d_{error}$ and A is the desired acceleration/deceleration.

The coefficient $d_{error}/|d_{error}|$ is then introduced to specify the sign of the desired speed variation since v_f is always positive and that the reference speed v_{ref} is given as the

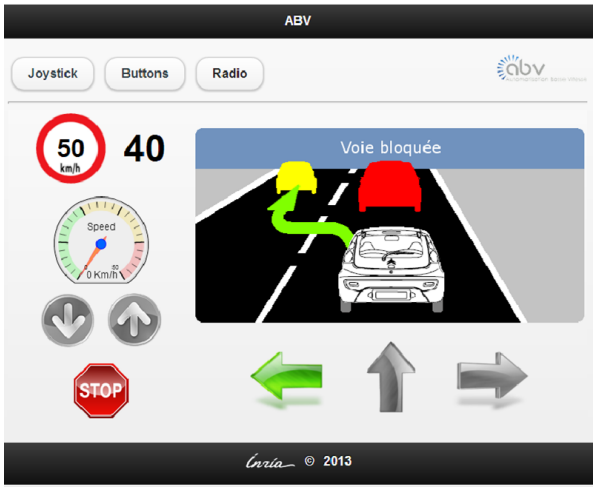

Fig. 9. Screenshot of the graphical interface in the interior of the vehicle

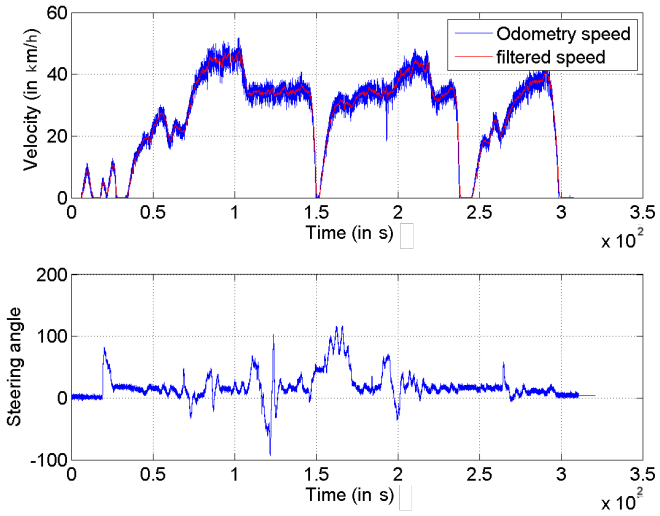

Fig. 10. Vehicle velocity

minimum value between $v_{desired}$ and the ACC speed v_{ACC} provided by:

$$v_{ACC} = v_{obs} + \left(\frac{d_{error}}{|d_{error}|}\right)\sqrt{2 \cdot A \cdot |d_{error}|} \qquad (16)$$

where d_{error} is given by:

$$d_{error} = \min\left(d_{obs}, v_{obs} \cdot t_{headway}\right) \qquad (17)$$

and d_{obs} is the distance from the vehicle to the obstacle, $t_{headway}$ is the headway time in seconds (2 seconds is considered to be a good value) and A is the desired vehicle acceleration if $d_{error} \geq 0$ or deceleration if $d_{error} < 0$.

The final gas/brake pedal position p is then given as the minimum between the previously calculated pedal position p using eq. 16 and an estimated collision risk multiplied by the maximum brake position. The Time-To-Collision (TTC) based collision risk estimation relates to the work described in [17].

IV. Interaction with a human

The human-machine interface in the vehicle consists of a touch screen allowing the driver to choose the preferred speed and to initiate lane changes by activating the blinkers if this is allowed in a given situation. Information about the current speed limits, environment situation (lanes and obstacles), and current vehicle maneuver are presented to the driver in real time by visual information in the screen. An image of the touch screen graphical interface is represented on Fig. 9. A situation awareness component monitors the current state of the lanes (number of lanes, lane marking type, current lane index), obstacles, ego vehicle state and current maneuver (keep lane, change lane left/right, emergency stop). It updates the information about the current situation to be presented in the graphical interface. Written messages about some particular situations (blocked road, end of application zone, emergency stop) are also used to inform the driver. At any moment the driver can bring the vehicle to a full stop by pressing the STOP button.

V. Experiment results

Real experiments have been conducted on the Satory tracks, close to Versailles in France. Results shown in Fig. 10 and 11 were registered during the final event of the ABV project. The demonstration consisted in driving the vehicle in a fully automated mode over 2 km. Weather was cloudy. The scenario satisfied a wide variety of road scenarios. For the lane marking detection, the algorithm has been confronted to damaged markings, useless blue road marking, paved road, various shadows,.... Additional tests have been made in the past to test the algorithm during night and that has been proved that the algorithm is even better int such conditions, where lane markings are really well reflected. Ego-localization had to deal with an outage due to the presence of trees. The experiment objective was to realize several use-cases: lane following, changing of lane, overtaking, ACC and emergency braking. All these maneuvers have been successfully conducted several times. Fig. 10 shows the vehicle velocity and its steering angle. The maximum velocity conducted by the vehicle was 47 km/h in a smooth trajectory as shown also in [18].

Fig. 11 represents the relative distance measured from the obstacle detection volume and the relative velocity calculated with the numerical derivator at the beginning of the scenario. When the scenario starts, a static obstacle is on the road. The vehicle slowly starts and stops behind the obstacle (at $t = 26$ s) Then, the obstacle starts (at $t = 32$ s) and the vehicle follows it, maintaining a safe distance according to its relative velocity. That one is calculated with the numerical derivator and keeps a smooth and coherent shape.

VI. Conclusion and future work

We presented in this article the prototype vehicle Citröen C1 Ev'ie and its technologies to demonstrate the technical feasibility of fully automated driving and low speed for the ABV project and this main objective was achieved.

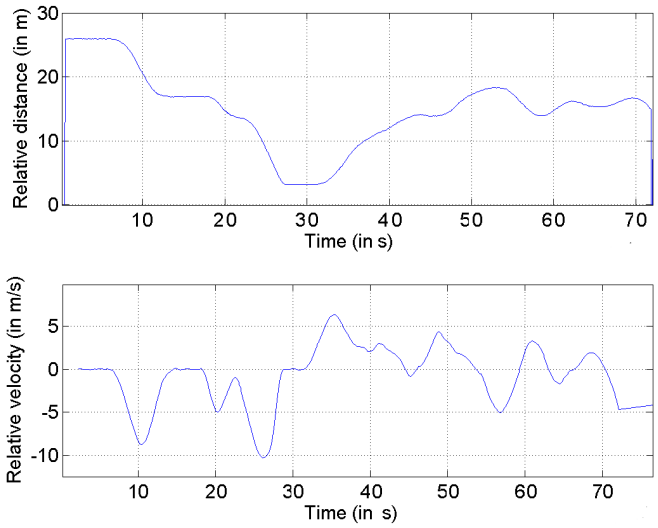

Fig. 11. Front obstacle relative distance and velocity

Now, beyond the ABV project objectives, many perspectives can be drawn. The perception system could be more robust including stereo-vision algorithm for the ego-localization and the obstacle detection algorithms. To complete the perception architecture to the level of the state of the art, localization should include a map matching algorithm. It could also integrate more functionality such as the road sign unit detection in order to be independent on the map. The decision system, quite simple for the ABV project, can be improved, in order to integrate more complex situations: passing maneuvers with multi-criteria decisions, intersection management including round-about ... long terms perspectives being to ensure automation outside the application zone Many works are still in progress on the interaction between the driver and the car. There is no doubt that future of automation aims towards an intelligent interaction, able to assess the driver state as well as the system state.

ACKNOWLEDGMENT

This work is part of ABV (Automatisation Basse Vitesse), a French research project for low speed automation. The authors would like to thank the Agence nationale de la recherche (ANR) in France for supporting this project for the support of the French automotive cluster Mov'eo.

REFERENCES

[1] M. Parent, "Advanced urban transport: Automation is on the way," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 9 –11, march-april 2007.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-e. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. V. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley : The Robot that Won the DARPA Grand Challenge," vol. 23, no. April, pp. 661–692, 2006.

[3] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[4] M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, S. Debattisti, R. Fedriga, M. Felisa *et al.*, "The vislab intercontinental autonomous challenge: 13,000 km, 3 months, no driver," in *Proc. 17th World Congress on ITS, Busan, South Korea*, 2010.

[5] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV)*, June, pp. 163–168.

[6] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 333–347, 2013.

[7] [Online]. Available: https://www.youtube.com/watch?v=xsV8P7X4gvw

[8] M. S. Grewal and A. P. Andrews, "Kalman filtering: Theory and practice," 1993.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] H. Li and F. Nashashibi, "Robust real-time lane detection based on lane mark segment features and general a priori knowledge," *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 812–817, Dec. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6181387

[11] F. Nashashibi and B. Bargeton, "Laser-Based Vehicles Tracking and Classification Using Occlusion Reasoning and Confidence Estimationr," *IEEE Intelligent Vehicles Symposium*, pp. 847–852, 2008.

[12] a. Mendes, L. Bento, and U. Nunes, "Multi-target detection and tracking with a laserscanner," *IEEE Intelligent Vehicles Symposium*, pp. 796–801, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1336486

[13] M. Fliess, C. Join, and H. Sira-Ramirez, "Non-linear estimation is easy," *International Journal of Modelling, Identification and Control*, vol. 4, no. 1, pp. 12–27, 2008.

[14] M. Fliess and H. Sira-Ramírez, "An algebraic framework for linear identification," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 9, no. 1, pp. 151–168, 2003.

[15] M. Mboup, C. Join, and M. Fliess, "Numerical differentiation with annihilators in noisy environment," *Numerical Algorithms*, vol. 50, no. 4, pp. 439–467, 2009.

[16] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet, "Accurate automatic guidance of an urban electric vehicle relying on a kinematic gps sensor," in *5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'04)*, 2004.

[17] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu, "Intersection safety using lidar and stereo vision sensors," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 863–869.

[18] [Online]. Available: http://youtu.be/DoM06ho7JC0